A number of large computational applications are being analyzed in order to assess their potential for decomposition into modules for distributed processing. The current applications are:

a) Programs which use heuristic methods in decision-making. Heuristic programs frequently employ recursive decomposition of problems into subsidiary problems which themselves may be suitable for distributed processing.

b) Programs which use multi-faceted databases to retrieve and abstract information. The process of intelligent data retrieval and analysis often depends on data or knowledge sources which are being maintained at geographically distributed processing sites.

c) Programs which acquire data from multiple, possibly dissimilar, sensors and attempt to reduce this data to simpler hypotheses.

Further candidates for analysis are:

d) Programs which solve large numerical problems, such as those found in image processing applications.

e) Operating system functions.

How the parts of those programs work together depends on the coupling required between their processes and on the facilities to support this coupling by the hardware. Parameters which describe the computations to be simulated include:

a) The computational kernel size: the cycle and memory demand of a computational unit between interprocessor reference requirements.

b) The computation definition message size: the amount of data required to transmit sufficient information to sufficient information to initiate a computational kernel.

c) The database size: the amount of data or program text required to sustain a computational kernel, and its availability and residence in the network.

The behaviour of the system can be varied through the adjustment of other parameters. These parameters may be set to reflect the architecture of specific hardware systems, or may be varied to obtain optimum performance. In addition to obvious parameters (as the number and power of the processors), we expect the following parameter types to be important in developing an understanding of the spectrum of distributed processor architectures:

a) Interconnection density. As the density decreases, the message delay and congestion increase. This parameter will provide a high level abstraction of multi-processor connectivity schemes. Geographical distribution will increase message delay and transmission const.

b) Computational locality. A high degree of locality (of database or procedural information in the network) will enhance the probability that relevant knowledge exists in closely linked nodes, thus counteracting the effects of a low interconnection density.

c) Database viscosity. A database, including the programs required to carry out the computations at a node, may be more or less fixed to one specific node. This therefore encourages the use of certain nodes for specific functions. Many current processor networks are completely rigid in this sense, and for these networks optimal initial program and database allocations may be determined. However, we hypothesize that a greater degree of dynamic resource allocation is desirable to cope with changing loads and in order to enhance reliability. For this reason this parameter needs to be included.

d) Redundancy. In order to assess the cost and benefits in terms of responsiveness and reliability, the redundancy of database and computations will also be made a parameter. In order to utilize the redundancy well, the computational resources (programs or data) which effect system performance most must be identifiable.

e) Error rate. In order to test the effectiveness of reliability strategies, node and communications channel failures will be simulated.

An important aspect of this model is that we intend to keep the abstractions at a sufficiently high level to allow analytic and intuitive verification of the model behaviour when applied to well understood computations. In the past several large applications have been mapped into specific parallel machines, but these results are not easily transferred to new architectures. The distributed processor systems now being built may have characteristics with unpredicted effects on system behaviour. We expect to be able to use the model to find potential bottlenecks, which then will define areas where extra design attention has a high payoff.

We do not intend to build hardware which is based literally on the abstract model. We hope to verify results obtained from the model using existing distributed processor systems and, assuming that our model (with appropriate parameters describing the load and architecture) matches the given system, be able to advise on system utilization or development aspects.

We are currently active in the support of one specific architecture, the S-1 Advanced Technology Processor. This system promises to deliver very high performance through the use of quite powerful processors nodes (on the level of the DEC KL-10 or more), large shared address spaces (30 bit addresses), and a moderate number of parallel processors (16 for the Mark II version). The fact that memory is shared allows the coupling to be quite strong, but partitioning of problems with less coupling will yet incur less overhead and interference.

The hardware for the prototype processor (Mark I) node is now beginning to be operational, basic system software has been written. The PASCAL language is now available, a FORTRAN compiler is being developed, and a LISP compiler is being considered as the next task. This project is funded through Lawrence Livermore Laboratory and supported by various agencies. While this implementation is only one of the possible architectural choices for multi-processors it is encouraging that we are obtaining realistic resources to verify and test notions of parallel processing.

B. Medical Relevance and Collaboration

Many applications at SUMEX consume large quantities of computational resources. They are also being hampered by architectural limitations of the current hardware, of which the 18 bit address is probably the worst constraint. The use of multiple distributed processors may provide a means to gain the required processing capabilities in an economic manner. In this sense the medical relevance of this study is indirect. We are attempting to develop tools which will be of use in medical computation problems.

Our studies in distributed database applications have a more direct medical relevance. To this end, we are maintaining contact with Dr. Jim Fries, whose ARAMIS database network collects data for the analysis of disease progress and treatment efficacy in rheumatoid arthritis from a variety of institutions. Sharing of data to provide a broader base for analysis is also a feature of programs in cardiology and oncology in which physicians at Stanford participate. We are also discussing with Dr. Collen and his collaborators at Kaiser Permanente the benefits, costs, and trade-offs of distributed computing. The development of adequate concepts and models is important to lay a bridge which supports communications between Medical and Computer Scientists.

In each of these instances the distributed nature of the data resources leads to differences in the meaning of data items, so that simple aggregation of the data may not be valid. Distributed processing may provide a powerful alternative. We see here flow of partially processed data, i.e., information, between the nodes, rather than row data of unspecified semantic value.

C. Progress Summary

The HYDROID project has been underway since the fall of 1976. We have been involved since that time in developing a basic understanding of important problem areas in distributed processing and problem solving.

A weekly research seminar, begun in December 1976 has brought together members of the faculty and students from a variety of disciplines, and has included several speakers from application areas where distributed processing may be beneficial. Recently this seminar has been combined with a research seminar on communication issues, and this quarter a joint course on the topic of distributed computing is being presented for the first time and has attracted about 150 student from Stanford and industry. Development of industrial participation is deemed important since the resources for these systems require joint efforts.

A formalism to express the control of distributed problem solving in loosely-coupled processor networks is the subject of the nearly completed Ph.D. thesis of Reid Smith. This CONTRACT NET protocol makes the interprocessor interactions explicit. It is the cost associated with these interactions which appears to generate one of the performance boundaries for distributed processor systems.

Problems of distributed search, distributed data acquisition, and distributed databases have been simulated, as well as a model which studied network topologies.

D. Publications

1) H. Garcia-Molina, "Overview and Bibliography of Distributed Databases", Computer Science Department, Heuristic Programming Project, Stanford University.

2) H. Garcia-Molina,"Distributed Database Coupling", Heuristic Programming Project, Paper HPP-78-4, Computer Science Department, March 1978.

3) H. Garcia-Molina, "Performance Comparison of Update Algorithms for Distributed Databases",Heuristic Programming Project, Paper HPP-78-6, March 1978.

4) K. Knudsen, "Some Issues in the Design of Large Multi-Microprocessor Networks", Stanford Heuristic Programming Project, Paper HPP-77-31.

5) R. G Smith, "A Formalism for Distributed Problem Solving", Proceedings of the Fifth International Joint Conference on Artificial Intelligence, MIT AI Lab, Cambridge, August 1977.

6) R. G. Smith, "Issues in Distributed Sensor Net Design", Heuristic Programming Project, HPP-78-2, January 1978.

7) R. G. Smith and R. Davis, "Distributed Problem Solving: The Contract Net Approach", to be presented at the 2nd National CSCSI Conference, Toronto, July 1978.

8) Thomas McWilliams, Lawrence C. Widdoes, Jr., and Lowell L. Wood, "Advanced Digital Processor Technology Base Development for Navy Applications; The S-1 Project",Lawrence Livermore Laboratory, Report NCID-17705, September 1977.

9) G. Wiederhold and I. Kuhn, "Effective Services of Automated Ambulatory Medical Record Systems", Policy Analysis and Information Systems, Vol. 1 2, January 1978, pp. 21-32.

10) G. Wiederhold, "Binding in Information Processing", in preparation.

11) G. Wiederhold and R. El Masri, "A Structural Model for Database Systems", submitted for publication.

12) G. Wiederhold, "Introducing Semantic Information into The Database Schema", Proceedings of the Annual Conference of the Canadian Information Processing Society, May 1978.

13) G. Wiederhold, "The Data Model of MUMPS Globals", Proceedings of the MUMPS Users Group Meeting, June 1978.

E. Funding Support Status

The HYDROID project is currently supported under the Heuristic Programming Project contract with the Advance Research Projects Agency of the DOD, contract number MDA 903-77-C-0322, E. A. Feigenbaum, Principle Investigator.

II. INTERACTIONS WITH SUMEX-AIM

SUMEX-AIM currently provides mainly the resources for communication and documentation for the project. Recently, simulations, although initially developed at SUMEX, have been carried out on IBM 370 computers at SCIP and SLAC at Stanford due to their very high demands on computing cycles. System work for the S-1 project is carried out at the SLAC and at the SAIL facilities. Further funded work on databases will use resources at SRI. The availability of the ARPA-net has made it possible to shift work to processors where resources and funding are most appropriate. We thus enjoy a high degree of interaction with other projects involved in the problems which result from construction of large programs. Other points of contact are related to the use of the same programming languages as well as the abundance of AI expertise residing around the resource. This latter point is especially important considering that one of our aims is discovery of suitable mappings of well understood AI methods onto highly parallel asynchronous processor networks.

SUMEX-AIM is also an excellent medium for informal transmission of reports, recent results and bulletins to users with related interests and problems. The powerful screen-oriented editors available greatly enhance our capabilities for writing both text and programs.

Finally, the development of simulation programs generally requires a highly interactive computing environment - the sort of environment that SUMEX-AIM provides.

4.2.5 MOLGEN PROJECT

MOLGEN - An Experiment Planning System for Molecular Genetics

Prof. J. Lederberg (Genetics, Stanford)
Prof. N. Martin (Computer Science, U. of New Mexico)
Prof. E. Feigenbaum (Computer Science, Stanford)

I. SUMMARY OF RESEARCH PROGRAM

A. Technical Goals

The MOLGEN project is developing a computer system capable of generating the experiment-planning sequences needed to solve given structural problems in molecular genetics. In particular we have developed a system which is capable of acquiring and representing information about genetic objects, transformations, and strategies. The knowledge base presently includes information on DNA structures, restriction enzymes, laboratory techniques, and a growing collection of genetic strategies for discovering information about various aspects of DNA molecules. Several specific subproblems such as simulating Ligase enzymes, determining safe restriction enzymes for gene excision, and inferring DNA structures from segmentation data have been explored. We have designed our effort to facilitate generalization to other domains beyond genetics in future research and applications.

The MOLGEN project has both an applications and a computer science dimension. Along the latter dimension, we seek to deepen our knowledge of the art and science of creating programs that reason with symbolic knowledge to aid human problem solvers. The task domain, molecular genetics, serves as a rich intellectual and scientific environment in which to develop and test our ideas.

The major computer science issues we are addressing during the current grant period are:

(1) Creation of a knowledge representation system with a knowledge acquisition package. The system, known as the Units Package, may be used to build a knowledge base in any suitable domain. It provides an object-centered approach for storage of both declarative and procedural information concerning all entities in the domain.

(2) Structured representation of process information. Procedures which simulate the action of the various processes in the domain form an integral part of the knowledge base. Moreover, the representation framework allows for inspection and acquisition of those procedures.

(3) Creation of program schemata and instances for general problem solving steps. Domain-independent knowledge about general problem solving methods also fits into the knowledge representation structure we have devised.

(4) Domain Specific Critics. Mechanisms for the activation of various domain specific strategies when certain predefined situations occur during the course of experiment design.

(5) Development of a specific planning strategy designed to provide high-performance for the class of genetic experiments known as discrimination experiments. The idea is based on indexing abstracted experimental designs to the types of structural features for which they have proven useful.

Along the applications dimension, we are attempting to develop tools that can benefit molecular geneticists. We believe there is substantial benefit to be derived from programs that act as "intelligent assistants" to scientists. First of all, the sheer amount of detailed knowledge a scientist is expected to know makes it likely that good experiments are being missed. Second, we believe that an intelligent planning assistant can offer some help in reasoning about the consequences of combining experimental facts in many possible ways.

A third motivation for applying artificial intelligence techniques to an experimental science like molecular genetics is to help us better understand the scientific method. The rigorous detail required for creating computer programs that assist in the performance of scientific tasks forces us to explicate concepts and procedures much more carefully than practicing scientists usually do.

B. Medical relevance and collaboration

Molecular genetics has at least two major connections to medical research. Learning about the basic mechanisms which control the operation and transmission of genetic information is necessary to understand and treat the wide range of diseases and health conditions that are genetically controlled. Also, recent developments in molecular genetics offer the promise of using genetic mechanisms to produce essentially limitless amounts of drugs ard other biomedical substances.

The MOLGEN project is a joint effort of the Computer Science Departments of Stanford and the University of New Mexico and the Genetics Department of Stanford. Major participants are Professor Nancy Martin and James Challenger of the University of New Mexico; Professor Edward Feigenbaum, Professor Bruce Buchanan, Dr. Randall Davis, Peter Friedland, and Mark Stefik of the Stanford Computer Science Department, Professor Joshua Lederberg, and Jerry Feitelson of the Stanford Genetics Department, and Professor Lawrence Kedes of the Stanford Medical School. Jim Case, a graduate student in Professor Douglas Wallace's laboratory, and Dr. John Sninsky, a molecular biologist working in Professor Stanley Cohen's laboratory, are also collaborating in the MOLGEN project.

C. Progress summary

The major effort in MOLGEN has been the creation of a knowledge management system. In addition, several specific problems which arise in genetics have been examined in sufficient detail to result in reports and/or special purpose programs. We report briefly on two such programs, SAFE written by Peter Friedland, and GA-1 written by Mark Stefik.

Knowledge management system

The success of MOLGEN as an experiment planner will depend on the quality of its knowledge base. Therefore, much of the research effort to date has been in the design and implementation of a knowledge representation and acquisition system. All of the information relevant to the planning process will be an explicit part of the knowledge base. The motivation for this aspect of the design is the necessity to expand the program capabilities in a modular fashion and to explain the rationale behind the program's planning behavior. We need to represent concepts (e.g. enzyme), instances (e.g. EcoRI), relationships among concepts, and relationships among instances. In addition, we need to represent processes. We have purposely limited the expressive power of our representations to enable us to clearly define their semantics.

The result of this work is the Unit Package. Although this package has been designed in the context of our genetics application, the package does not contain any genetics knowledge.

One important aspect of the design of the system is that the knowledge base contains knowledge about its own data representations. We have provided what we term a "bootstrap knowledge base." It contains domain independent knowledge about commonly used data types. When using our knowledge base in a new domain, an artificial intelligence researcher would probably start with the bootstrap knowledge base and then proceed to create units for the specific knowledge of his task area. Both the AGE and genetics knowledge bases have been started in this manner. The bootstrap knowledge base serves to illustrate our approach to extensibility. Most of the bootstrap knowledge base is made up of primitive datatypes. To add a new datatype to our system, one needs to provide the knowledge base with procedures for some basic operations -- such as editing and printing. Actually, the same approach is used in the unit package for defining a new datatype as is used for defining a new enzyme. The process of defining new datatypes requires, however, an understanding of Interlisp because the primitive processes in the system are grounded in that language. New datatypes must be defined together with their basic operations and entered into the knowledge base.

Knowledge base contents

The genetics knowledge base is growing rapidly. Approximately 60% of the commonly used enzymes have been characterized. A beginning has been made on the characterization of organisms such as bacteria and phages, plasmids and other vectors, and genes. Our knowledge base also contains a growing collection of genetic strategies for discovering information about various aspects of DNA molecules, as well as a hierarchy of laboratory techniques which are used to instantiate the strategies. The hierarchy of techniques includes modification, separation, visualization, sequence analysis, and bacteriological techniques at many levels of abstraction.

Safe program

The geneticist needs to predict what restriction enzymes can safely be used to excise a gene, i.e. which ones can be guaranteed not to cut the functional part of the gene. We would also like to know the approximate location of the possible cutting sites of other restriction enzymes. This would all be very easy if the complete DNA sequence of the gene was known. Sequence information is becoming more and more prevalent, but it is still uncommon to know the complete sequence of a gene. However, it is not unusual to know what protein the gene codes for and to know the amino acid sequence of a protein.

Knowing the amino acid sequence does not provide full information because of the degeneracy in the genetic code. One codon (a triplet of nucleotides) specifies only one amino acid, but up to six different codons may specify the same amino acid. The problem therefore, is combinatorically difficult. Typical proteins are up to 300 amino acids long (900 nucleotides), and all possible nucleotide sequences which would produce the protein in the three possible phases have to be considered.

The SAFE program lists the restriction enzymes that are currently stored in the knowledge base and allows the user to add new ones. Besides determining which enzymes are safe to use for gene excision in a particular DNA molecule, the program also gives the position in the amino acid sequence where the possible cutting site would be located.

GA-1 program

A common task in molecular genetics laboratories is the analysis of DNA structure from restriction enzyme segmentation data. This task is one of the simplest, although time-consuming, analysis tasks in molecular genetics. Two standard approaches to solving this problem were examined: a data-driven strategy and a model-driven strategy. These approaches are discussed and compared in terms of sensitivity to missing data, efficiency in the use of data, and other measures of performance in [Stefik, 78]. A program was designed and implemented which is superior to human performance on smaller problems, both in speed and reliability. However, on large problems human problem solvers can use extra structural constraints to out perform the program. The current program uses only constraints derived from segmentation data itself. Geneticists usually know additional information -- eg. that a given segment is on the end or that certain segments must be adjacent.

A further benefit of this work was the suggestion of two new lab techniques: combining multiple enzyme digests and incomplete digests. These ideas arose from a systematic examination of evidence and inference rules that went into building the program.

D. Publications

Challenger J., A Program for Printing DNA Structures, CIS Report 78-3 (April 1978)

Feitelson J., Stefik M.J., A Case Study of the Reasoning in a Genetics Experiment, Heuristic Programming Project Report HPP-77-18 (Working Paper) (May 1977)

Martin N., Friedland P., King J., Stefik M.J., Knowledge Base Management for Experiment Planning in Molecular Genetics, Fifth International Joint Conference on Artificial Intelligence. 882-887 (August 1977)

Stefik M., Friedland P., Machine Inference for Molecular Genetics: Methods and Applications, Proceedings of the National Computer Conference, (June 1978)

Stefik M.J., Martin N., A Review of Knowledge Based Problem Solving As a Basis for a Genetics Experiment Designing System, Stanford Computer Science Department Report STAN-CS-77-596. (March 1977)

Stefik M., Inferring DNA Structures From Segmentation Data: A Case Study, Heuristic Programming Project Report HPP-78-3 (January 1978)

E. Funding Support Status

MOLGEN is funded by two NSF grants for the 2-year period 7/1/76 through 7/1/78. These grants, including indirect costs, are:

MCS-7611935	PI:	Nancy Martin, UNM Univ. of New Mexico	Amount of $68,000
MCS-7611649	PI:	Edward Feigenbaum Joshua Lederberg, Stanford University	Amount of $110,700

Two additional grants have been requested for the period 8/1/78 through 8/1/80.

II. Interactions With The SUMEX-AIM Resource

All system development has taken place on the SUMEX-AIM facility. The facility has not only provided excellent support for our programming efforts but has served as a major communication link among members of the project. Through the SUMEX-AIM facility, program development has taken place concurrently at Stanford and New Mexico. Systems available on SUMEX-AIM such as INTERLISP, TV-EDIT, and BULLETIN BOARD have made possible the project's programming, documentation and communication efforts. The interactive environment of the facility is especially important in this type of project development.

We have taken advantage of the collective expertise on medically-oriented knowledge-based systems of the other SUMEX-AIM projects. In addition to especially close ties with other projects at Stanford, we have greatly benefitted by interaction with other projects at yearly meetings and through exchange of working papers and ideas over the system.

The combination of the excellent computing facilities and the instant communication with a large number of experts in this field has been a determining factor in the success of the MOLGEN project.

III. Research Plans

A. Project goals and plans

In exploring the three major motivations mentioned in section I.A. for creating the MOLGEN project, there are many specific subproblems. We have identified five for concentrated effort in the next two year period.

(1) Creating a more comprehensive genetics knowledge base. Expanding the knowledge base within the area of DNA structural manipulation problems.

(2) Abstracting and Saving Plans. Recognizing when newly-created experiment designs are worth saving and then generalizing those plans so they are useful for more than the specific problem environment which caused their generation.

(3) Making use of the process of hypothesis formation to help debug MOLGEN-produced experiment designs. This process is especially important in a domain like molecular genetics where incomplete knowledge about objects and processes is the rule rather than the exception.

(4) Experiment planning by analogy. MOLGEN provides an excellent environment for exploring various types of analogical reasoning. We integrate problem-solving by analogy into the experiment design system as one of the possible tools for solving subproblems.

(5) Performance evaluation as an integral part of the knowledge representation and acquisition system. We view the process of evaluating a system's performance and suggesting improvements as an AI problem solving task. The strategies for this evaluation will be stored within the same framework as all other MOLGEN planning strategies.

Each time the Heuristic Programming Project at Stanford has built another large AI program, we have learned more about how to do it better and faster next time. For example, the production rule interpreter in Heuristic Dendral (for special-purpose rules) became the general rule interpreter of MYCIN. One of the significant products of MOLGEN research will be the sets of ideas and programs for encoding and manipulating large amounts of knowledge about a scientific discipline. We have transferred some parts of the MOLGEN Units package to another project interested in building a knowledge base about AI methods and techniques. Making the tools used here available for use in new programs is an important aspect of our work and is generally important for cumulation of knowledge in the AI field. In order to do this we must reformulate the methods so they are more generally applicable and more readily combined in diverse ways.

B. Justification and requirements for continued SUMEX use.

The MOLGEN project is completely dependent on the SUMEX facility. While we have solved many of the original problems facing us in a manner useful to working geneticists, we are just in the middle phase of building a planning system. Without support from SUMEX to complete this system, many of the results of the last two years will be ineffective. In the past six months our interactions with geneticists outside of Professor Lederberg's laboratory have increased greatly. The geneticists are excited about helping us with our knowledge base. Also, with our help, they are finding useful ways to use the computing facility in their current research. Thus the serendipity of supporting MOLGEN is the creation of many useful computer research programs.

We are asked to state our requirements for continued SUMEX use. We project that our usage of processor cycles and file storage will grow to twice the current levels in the coming year.

4.2.6 MYCIN PROJECT

MYCIN - Computer-based Consultation in Clinical Therapeutics

S. N. Cohen, M.D. (Genetics and Pharmacology) and
B. G. Buchanan, Ph.D. (Computer Science)
Stanford University

I) Summary of research

The MYCIN program is an outgrowth of nearly a decade of work on DENDRAL. We are building on, and improving, many of the ideas from DENDRAL about representing large amounts of domain-specific knowledge for computer aided problem solving. The representation, use and acquisition of knowledge for computer programs has been called "knowledge engineering" [D.Michie, On Machine Intelligence, New York: Wiley, 1974]. We represent that body of knowledge as a collection of decision rules about diagnosis and therapy selection in infectious diseases. These conditional sentences are called "production rules". The production rule formalism provides an easily understood representation of facts and relations.

The MYCIN knowledge base currently consists of approximately 400 such rules. Each rule consists of a set of preconditions (called the "premise") which, if true, justifies the conclusion made in the "action" part of the rule (an example is shown below).

```
If     1) the gram stain of the organism is gram negative,
   and 2) the morphology of the organism is rod,
   and 3) the aerobicity of the organism is anaerobic,
Then there is suggestive evidence (.6) that the identity of
   the organism is Bacteroides.
```

The Medical Problem

A number of recent studies (documented in Shortliffe, 1976) indicate a major need to improve the quality of antimicrobial therapy. Almost one-half of the total cost of drugs spent in treating hospitalized patients is spent on antibiotics, and if results of a number of recent studies are to be believed, a significant part of this therapy is associated with serious misuse. Some of the inappropriate therapy involves incorrect selection of a therapeutic regimen, while another serious problem is the incorrect decision to administer any antibiotic. One recent study concluded that one out of every four people in the United States was given penicillin during a recent year, and nearly 90% of these prescriptions were unnecessary. Other studies have shown that physicians will often reach therapeutic decisions that differ significantly from the decisions that would have been suggested by experts in infectious disease therapy practicing at the same institution.

Initial culture reports from a microbiological laboratory may become available within 12 hours from the time a clinical specimen is obtained from the patient. While the information in these early reports often serves to classify the organism in general terms, it does not often permit precise identification. It may be clinically unwise to postpone therapy until such identification can be made with certainty, a process that usually requires 24 to 48 hours, or longer. Thus it is commonly necessary for the physician to estimate the range of possible infecting organisms, and to start appropriate therapy even before the laboratory is able to identify the offending organism and its antibiotic sensitivities. In this setting MYCIN plays two roles: (a) providing consultative advice that will assist the physician in making the best therapeutic decision that can be made on the basis of available information, and, (b) by its questioning of the physician, pinpointing the items of clinical data that are necessary to increase the validity of the clinical decision.

Progress Summary

In the past year our work has been guided by three fundamental objectives, roughly corresponding to use, explanation, and acquisition of medical knowledge in a symbolic reasoning program. These are discussed in the next three sections.

Use of Knowledge

A major objective of the MYCIN system has been to provide a computer-based therapeutic tool designed to be useful in both clinical and research environments. This requires development of a system that has a medically and scientifically sound knowledge base, and that displays a high level of competence in its field.

Expansion of Knowledge Base

Work on improvements to the knowledge base included both expansion into new areas and further development of existing areas of expertise. A start has been made, for instance, on rules for dealing with cystitis infections, a very common source of bacterial infection. In addition, a urinary tract infection therapy grid has been developed. This is a large table containing basic information about drug and dose selection, which functions as a basic information source for the therapy selection routines in the program.

We have also developed a facility that plots the steady state blood levels of antibiotics, based on a range of patient-specific parameters (body surface area, level of kidney function, etc.). This facility presents a very clear picture of the consequences of selecting various dose levels and intervals, and will be very useful in helping a clinician select a therapy regimen that is maximally effective without endangering the patient due to toxic levels of drug in the blood.

Extensions to the system included improving the design of rules so that they now include "justifications" and literature references. That is, rather than simply indicating that "gram negative rods in a nosocomial infection are likely to be pseudomonas", the rule now has a "justification" which is a further explanation of the reasoning behind the rule. This justification can be printed out at the users request, and may indicate, in this case, that pseudomonas is more common in the hospital setting, and hence a likely causative organism in nosocomial infections. The literature references are pointers to published articles that give further information (including original clinical studies) which serves to illustrate the reasoning behind the rule.

Finally, as a result of our recent evaluation of the system's performance on meningitis cases, we have made several improvements to the body of rules dealing with this area.

Evaluation of Competence

A formal evaluation of Mycin's performance in recommending treatment for patients with infectious meningitis was begun in May of 1977. The study design enabled unbiased comparisons to be made of Mycin's therapy recommendations with those of infectious diseases specialists and physicians of varying degrees of medical expertise. These comparisons provided a means to establish the level of competence of the Mycin system in prescribing therapy for patients with meningitis.

Ten patients with infectious meningitis of variable etiologies were selected for the study by a physician not familiar with the Mycin system. A detailed clinical summary was compiled for each patient. These summaries were presented to five faculty members in the Division of Infectious Diseases at Stanford, one senior research fellow in infectious diseases, a senior resident in medicine and a senior medical student, with a request to select therapy for each case. A Mycin consultation was performed for each patient using the same clinical summary data. As a result, a total of ten therapy recommendations, including the actual therapy the patient received, were compiled for each case.

The patient summaries were then given to eight prominent infectious diseases specialists at institutions other than Stanford. They were asked to select therapy for the patients and to evaluate the recommendations of the Stanford physicians, the student, Mycin and the actual therapy the patient received. The therapies were listed in random order for each case and the national evaluators were not informed of the identities of the ten prescribers. The evaluators used the following criteria to rate the prescribers' recommendations: "equivalent", the recommendation was identical or equivalent to his own; "close call", the recommendation was different but the evaluator was willing to rate it as an acceptable alternative; and "not acceptable", the recommendation was inappropriate.

The results of the evaluation, as summarized in the table below, demonstrate that Mycin's competence in selecting antimicrobial therapy for meningitis is comparable to that of the infectious disease faculty at Stanford.

Prescribers	Acceptability Rating
MYCIN	52
Faculty-1	50
Faculty-2	48
ID Fellow	48
Faculty-3	46
Actual therapy	46
Faculty-4	44
Resident	36
Faculty-5	34
Student	24

The acceptability rating is the cumulative total of therapy selections rated as "equivalent" or a "close call". Since there are eight national evaluators and ten cases to be evaluated a perfect score would have been 80. Mycin's therapy recommendations received the largest number of acceptable ratings.

Use of Prototypes to Guide the Reasoning

A consultation program is currently under development which uses a set of domain-specific prototypes of typical situations to guide the invocation of production rules. This program has been implemented in the domain of pulmonary physiology in which the prototypes represent the various pulmonary diseases. Each prototype consists of a group of situation-specific components, each having plausible values, likely trap values and a possible default value.

The addition of prototypes to the rule-based system allows us to:

1) Control the consultation using information given in the prototypes. The determination of values of parameters in the original consultation system (which in turn causes rules to be tried and questions to be asked) is replaced by the determination of values of components in a prototype, i.e., "filling in" the prototype. This "search" for information is more focused since it occurs within a more limited context, namely, the context defined by the prototype.

   In addition, once a prototype is chosen, a "control element" associated with the prototype is invoked. This control element explicitly states actions to be performed in "filling out" the prototype.

2) Analyze the result of the consultation when a value is unknown vs. the result when a default value is used. That is, if we provide a default value for a component, does it make a difference in the result? For example, if it is found that using a default value for a missing lab test results in a different therapy for a patient, then the user can be informed of the significance of that lab test.

3) Detect inconsistencies in the data using the Plausible Values associated with the component in the prototype.

4) Detect error conditions and suggest actions to correct them using the Likely Trap values associated with the component in the prototype.

5) Give explanations of system performance in terms of the prototypes. A trace of system actions is provided by noting which prototype is invoked at any time.

6) Classify consultations in terms of the prototypes, allowing indexing and retrieval of consultations for uses such as testing a new rule relevant to some situation.

Compilation of the Rule Base into a Decision Tree

In an effort to overcome the inherent slowness of interpreting an increasingly large set of production rules, work is underway to "compile" the rule base. The block of rules which make conclusions about a particular clinical parameter implicitly constitutes a program to conclude the parameter. Under the present control structure, that program is trivial; the rules are simply interpreted one at a time; if a predicate appears in 50 rules, it will be tested 50 times. The rule compiler, however, can construct an explicit LISP program from those rules, and, by making compile-time inferences, restructure it to eliminate redundant computation of related premise clauses. The result is an optimal decision tree, which effectively tries several rules in parallel; a single test in the compiled program (e.g. "what is the infection?") can cause a large set of rules to fail at once.

The rule compiler will thus allow the consultation program to use an efficient deductive mechanism, even as the rule base expands, while the flexible rule format is still available for explanation and debugging.

Development of a Tutoring System

The objective of this part of the project is to implement and test a computer program for tutoring medical students and physicians in infectious disease diagnosis and therapy. The problem-solving and dialogue capabilities of this tutor are distinct so that arbitrarily complex case histories may be discussed with the student or physician.

The tutor itself has the capability to discuss the problem and the expert program's solution with a student. Therefore, the teaching capability is distinct from the problem-solving expertise of the consultation program. This will enable us to study alternative teaching methods.

This work proceeds by formalizing the conventional rhetorical patterns one finds in tutoring dialogues. The resultant formalization is, as in the consultation program, independent of any particular case, so the tutor will have the capability to discuss any case presented by a student or selected from a computerized library of cases. This project is similar in style to current research in the area of "intelligent computer aided instruction." The power of these programs derives from 1) separation of problem solving ability from teaching strategies, 2) construction of a student model which describes the student's understanding with respect to the decisions of the expert program, and 3) consideration of explanation techniques that make a point clearly and capture what the student needs to know.

The initial design of the tutor has involved three major steps:

1) Modification of the control structure of the expert program to leave behind detailed traces for use by the tutorial program. These traces indicate specifically which information was necessary to apply each rule; they are significantly more complex than the traces used by the current question-answer program in MYCIN.

2) Formalization of tutorial discourse procedures. Proceeding from an extensive hand simulation, patterns in the tutorial dialogue were formalized into procedures that constitute a sequence of action options for the tutor. For example, the most complex procedure details how to discuss a topic with the student: it captures the exchange of initiative as the student requests more data and the tutor quizzes him about new conclusions he can draw. Another procedure captures the situation in which the tutor mentions rules that are related to one that the tutor and student have just discussed. These procedures have been written in stylized INTERLISP so they are translatable into an English form. This makes it easy for us to show other researchers and educators the specific rhetoric patterns, domain knowledge, and teaching strategies used by our program. There are about 20 such procedures in the current formalization of the case method dialogue.

3) Formalization of teaching strategies and the communication model. Again using stylized code, rules have been written to decide what the student knows during the dialogue, based on his behavior and past history. For example, given a student hypothesis, the tutor applies communication model rules to determine which of the expert (domain) rules have probably been considered by the student. The teaching strategies use this belief as a starting point and make an appropriate response. A range of strategies have been formulated, consisting mainly of ways to state, hint or quiz about a rule. A taxonomy of question types based on difficulty allows the tutor to choose a strategy appropriate to the student's knowledge of the (domain) rule. Thus, the communication model and teaching strategies serve to control the rhetorical options that have been specified in the discourse procedures.

4) Augmentation of the expert knowledge base. Several forms of meta-level knowledge and annotations have been added to the (domain) knowledge base. These include: a) rule schema, or templates that describe a type of rule and its import (entered by a domain expert); b) key factors, or distinguished rule clauses that are derived by the tutor from the rule schema and indicate the key piece of information being taken into account by the rule; and c) factor interrelationships, such as an indication that when one factor is known other factors are not relevant.

Explanation of Reasoning

One of the important contributions of the MYCIN work to artificial intelligence has been our persistent emphasis on an intelligent system with the ability to explain the reasoning behind its decisions. It should be able to do so in terms that suggest to the physician that the program approaches the problem in much the same way that he does. This permits the user to validate the program's reasoning, and modify (or reject) the advice if he believes that some step in the decision process is not justified. It also gives the program an inherent instructional capability that allows the physician to learn from each consultation session.

The use of a rule-based representation of knowledge makes it possible for the system to explain the basis for its recommendations. For example, if asked "How did you determine the identity of the organism?" the program answers by displaying the rules which were actually used, and explaining, if requested, how each of the premises of the rules was established. This is something which people readily understand, and it provides a far more comprehensible and acceptable explanation than would be possible if the program were to use a simple statistical approach to diagnosis.

Extension of Question Types

Additions to MYCIN's question-answering system within the last year fall into three categories: general improvements to the parsing mechanism, refinements to answers that are given, and new types of questions that can be recognized and answered.

The first major improvement to the parsing mechanism is that each word in a particular question is given a single interpretation. In particular, a word may not implicate both a context and a parameter, or more than one parameter. When dictionary pointers alone are not enough to distinguish among ambiguous interpretations of a word, the routine that generates an answer to the question will select the interpretation that best answers the question. For example, in a rule-retrieval question where a single word implicates several different parameters, the parameter that is used in the most rules will be chosen.

Another change is that it is no longer necessary to explicitly mention the name of some context in a question that is specific to the consultation. Rough semantic analysis is often successful in determining that a question is about the consultation, and in selecting the appropriate context.

A final improvement to interpreting the question is in the analysis of questions that mention more than one context from the consultation. In the past, one single context was chosen, and the selection was often arbitrary. Now, heuristics are used to choose among the contexts, and in certain questions, more than one will be used (e.g. in "DID YOU USE THE SITE OF CULTURE-1 TO DETERMINE THE IDENTITY OF ORGANISM-2?").

Questions that the system has always been able to answer now get answers that are closer to what the user wants. One example of this is that the system now differentiates between the conjunction and the disjunction of parameters in a question. In the past "HOW DO YOU USE THE FACT THAT AN ORGANISM IS A GRAM NEGATIVE ROD" would have been answered by listing the rules that use either negative gramstain or morphology rod instead of those that use both. In addition to simply listing the rules, the system now also summarizes how they are used (in the question above, it would add that most of the rules that use those two factors (gram negative, and rod) are used to conclude about the identity of the organism). When printing rules in answer to a question, the system now includes a brief explanation of when the rule will be used, and gives the user the option of seeing the rule's author, medical justification, and relevant literature references.

Some of the "holes" in the question-answerer's repertoire have been filled in. The system can now explain how it calculated the dosage of the drugs that it prescribed and it can explain how it decided to treat for particular infections and organisms.

Development of Automatic Testing System

We have added a mechanism for automatically testing the question-answering system which will be used to debug the system and to maintain it in a working state as the consultation system changes around it. Testing in the past has been tedious due to the necessary interaction with a person asking questions and verifying the validity of the answers.

The new testing program consists of two phases. The first allows a MYCIN expert to run QA and save information on sample questions that were answered correctly. The information includes the question itself, intermediate parses, and an encoding of the answer that was given. This information is stored in a file.

The second phase is undertaken after substantive changes have been made to the system. A batch job is submitted which will run a consultation, then ask the questions that were stored on the file created during the first phase of testing. This job writes a report of the comparison of the new parse and answer with what is stored in the file. The discrepancies that are noted in these reports can be used to pinpoint the sources of bugs in the QA system, if any.

Acquisition of Knowledge

A third major objective has been to provide the program with capabilities that enable augmentation or modification of the knowledge base by experts in infectious disease therapy, in order to codify knowledge in the domain, as well as to improve the validity of future consultations. The system therefore requires some capability for acquiring knowledge by interacting with experts in the field, and for incorporating this knowledge into its knowledge base.

Substantial effort was placed in updating sections of Teiresias, the knowledge acquisition system developed by R. Davis [Davis, 1976]. In particular the post-consultation review procedure was modified to utilize the existing question-answering facilities to describe how particular conclusions were made. This allows the user direct access to the full capabilities of the question-answering system while the system aids the expert in isolating the problems in the reasoning process by unfolding the reasoning tree using the methods of the QA module to describe conclusions of rules.

The actual rule-acquisition procedure has been improved by the development of a new parser to perform the actual translation of English to the internal LISP representation.

During the design and implementation of another knowledge base for an EMYCIN system, a number of issues dealing with the initial acquisition of a knowledge base were identified. They are primarily the result of having to elicit and encode substantial amounts of domain-specific knowledge from an expert who is a novice at explicating and representing this knowledge in the form of rules and parameters. We are pursuing the design of facilities which will aid the new expert during the initial stages of this knowledge explication process.

MYCIN PUBLICATIONS

Shortliffe, E H, Axline, S G, Buchanan, B G, Merigan, T C, Cohen, S N. An artificial intelligence program to advise physicians regarding antimicrobial therapy, Computers and Biomedical Research, 6:544-560 (1973).

Shortliffe, E H, Axline, S G, Buchanan, B G, Cohen, S N, Design considerations for a program to provide consultations in clinical therapeutics, Presented at San Diego Biomedical Symposium 1974 (February 6-8, 1974).

Shortliffe, E H, MYCIN: A rule-based computer program for advising physicians regarding Antimicrobial therapy selection, Thesis: Ph.D. in Medical Information Sciences, Stanford University, Stanford CA, 409 pages, October 1974. Also, Computer-Based Medical Consultations: MYCIN, American Elsevier, New York, 1976.

Shortliffe E H MYCIN: A rule-based computer program for advising physicians regarding antimicrobial therapy selection (abstract only Proceedings of the ACM National Congress (SIGBIO Session), p. 739, November 1974. Reproduced in Computing Reviews 16:331 (1975).

Shortliffe E H, Rhame F S, Axline S G, Cohen S N, Buchanan B G, Davis R, Scott A C, Chavez-Pardo R, and van Melle W J MYCIN: A computer program providing antimicrobial therapy recommendations (abstract only). Presented at the 28th Annual Meeting, Western Society For Clinical Research, Carmel, CA, 6 Feb 1975. Clin. Res. 23:107a (1975). Reproduced in Clinical Medicine, p. 34, August 1975.

Shortliffe, E H and Buchanan, B G, A Model of Inexact Reasoning in Medicine, Mathematical Biosciences 23:351-379, 1975.

Shortliffe, E H, Davis, R, Axline, S G, Buchanan, B G, Green, C C, Cohen, S N, Computer-based consultations in clinical therapeutics: explanation and rule acquisition capabilities of the MYCIN system, Computers and Biomedical Research, 8:303-320 (August 1975).

Shortliffe E H, Axline S, Buchanan B G, Davis R, Cohen S, A computer-based approach to the promotion of rational clinical use of antimicrobials, International Symposium on Clinical Pharmacy and Clinical Pharmacology, Sept 1975, Boston, Mass. (invited paper)

Shortliffe E H, Judgmental knowledge as a basis for computer-assisted clinical decision making, Proceedings of the 1975 International Conference on Cybernetics and Society, pp 256-7, September 1975.

Davis R, King J J, An Overview of Production Systems, Machine Intelligence 8: Machine Representations of Knowledge (eds E W Elcock and D Michie), John Wylie, April 1977.

Davis R, Buchanan B G, Shortliffe E H, Production rules as a representation for a knowledge-based consultation system, Artificial Intelligence, Vol 8, No 1 (February 1977).

Shortliffe E H, Davis R, Some considerations for the implementation of knowledge-based expert systems, SIGART Newsletter, 55:9-12, December 1975.

Scott A C, Clancey W, Davis R, Shortliffe E H, Explanation capabilities of knowledge based production systems, American Journal of Computational Linguistics, Microfiche 62, 1977.

Wraith S, Aikins J, Buchanan B G, Clancy W, Davis R, Fagan L, Scott A C, van Melle W, Yu V, Axline S, Cohen S, Computerized consultation system for the selection of antimicrobial therapy. American Journal of Hospital Pharmacy 33:1304-1308 (December 1976).

B.G. Buchanan, R. Davis, V. Yu and S. Cohen, "Rule Based Medical Decision Making by Computer," Proceedings of MEDINFO.77, 1977).

Davis R, Applications of Meta Level Knowledge to the Construction, Maintenance, and Use of Large Knowledge Bases. Memo HPP-76-7, Stanford University, June 1976.

Davis R, Meta rules: content directed invocation, to appear, Proc ACM Conf. on AI and Programming Languages, August 1977.

Davis R, Knowledge acquisition in rule-based systems: knowledge about representations as a basis for system construction and maintenance, to appear, Proc. Conf. on Pattern-directed Inference Systems, May 1977.

Davis R, Interactive transfer of expertise: acquisition of new inference rules, to appear, Proc. Fifth IJCAI, August 1977.

Davis R, A decision support system for medical diagnosis and therapy selection, in "Data Base" (SIGBDP Newsletter), 8:58 (Winter 1977).

Davis R, Buchanan B G, Meta level knowledge: overview and applications, to appear, Proc. Fifth IJCAI, August 1977.

Clancey, W, The Structure of A Case Method Dialogue, accepted for publication in The International Journal of Man-Machine Studies, fall '78.

FUNDING

KNOWLEDGE-BASED INTELLIGENT SYSTEMS
THE MYCIN PROJECT

Bruce G. Buchanan, Principal Investigator
Adjunct Professor, Computer Science
Stanford University

National Science Foundation
Grant MCS77-02712

Budget for Year 1 from 6/1/77 through 5/30/78
Total Direct Costs $ 32,357.

2-Year Budget Summary from 6/1/77 through 5/30/79
2-Year Total Direct Costs 64,396.

II) Interactions with the SUMEX-AIM Resource

Collaborative Efforts

University of Rochester

Professor Charles Odoroff again requested access to MYCIN to use in his medical school class on medical computing. In the past, the students who examined MYCIN have provided valuable criticisms on which further developments can be based.

Carnegie-Mellon University

Dr. John McDermott, working with Professor Allan Newell, visited Stanford and has had extensive network communication with us about MYCIN's production rule formalism. He is translating MYCIN's infectious disease rules into their own OPS formalism in order to study the advantages and disadvantages of writing "fine grained"" rules, as in OPS, or "coarse grained" rules, as in MYCIN.

ISI

Drs. William Mann and James Moore of ISI visited Stanford to discuss, in part, their use of MYCIN to study man-machine communication. Because MYCIN has been developed for physicians to interact easily with the program , it was thought to be one of the most appropriate objects for their study.

EMYCIN

Part of our concern is to generalize the methods used in MYCIN, and make them available to others. We have developed a prototype of an "empty MYCIN" consultation system, called EMYCIN, that embodies most of the control structure, and none of the specific medical knowledge, of MYCIN. That is, for domains that are structured similarly to MYCIN's primary domain, the existing mechanisms for offering consultations can be coupled with knowledge (rules) of the new domain. Substantial progress has been made on providing a clear, consistent package for use by other groups.

Pulmonary Function Testing and Intensive Care Unit Monitoring

The MYCIN program and techniques have been adapted for two new medical/Artificial Intelligence applications. The first domain is the interpretation of the measurements made in the Pulmonary Function Laboratory from the Pacific Medical Center at San Francisco. This application uses the MYCIN program directly, but with a new rule set designed to diagnose the type and severity of pulmonary disease, and some measure of the reliability of the tests used in the laboratory. The second domain is the continuous interpretation problem for patients with mechanical breathing assistance in the Intensive Care Unit. This program uses the basic MYCIN architecture and concepts with extensions to handle the change in patient condition from time to time. Both of these applications are explained in more detail in the section on the PUFF project.

HEADMED

We have continued strong collaboration with Dr. J.Heiser at U.C. Irvine who has developed the HEADMED program from MYCIN. See HEADMED section for details.

Medical Consultation in Rheumatology

Preliminary design of a consultant for rheumatologic diseases was begun by Dr. R. Blum. This system would arrive at a diagnostic conceptualization of a case using a production rule type formalism as is used by MYCIN; however, fundamental revisions to the control structure would guide the rule invocation. It is anticipated that rule invocation would be in a data-driven (forward) direction rather than MYCIN's current goal-driven method. Rules would be embedded in a hierarchy of diagnostic hypotheses, conceptually similar to the INTERNIST Project at the University of Pittsburgh.

The presence of a large clinical databank would permit the addition of quantitative methods not presently utilized in the MYCIN formalism. The ARAMIS databank, a collection of clinical data on thousands of patients with rheumatologic diagnoses, will be used.

The major advantages of a consultant system compromised of both knowledge derived from clinicians and from a clinical databank would be 1) the capability for comparison of the two sources of knowledge, 2) the advantage over a pure databank approach of being able to provide advice when the data alone are insufficient, and 3) the advantage over a purely rule-based system of possessing quantitative precision when the data allow it in determining probabilities of symptom/disease association or in predicting outcome parameters.

DENDRAL

Many problems of developing complex reasoning programs in INTERLISP are common to both DENDRAL and MYCIN. We have continued close association with the DENDRAL project and are jointly working on solutions to the problems of managing large programs.

Critique of Resource Management

Management of this resource remains as professional as ever. The service from the staff, as well as from the computer, is superb.

As the computer becomes more heavily loaded we encounter more and more reluctance from physicians to use the program more than once because of the response delays. This inhibits our progress, needless to say. We are working on software modifications to alleviate the problem; we are also looking forward to acquisition of new system hardware, specifically the DEC 2020, to provide some relief.

III. Research Plans

Long Range Goals

We intend to build the MYCIN rule-based consultation program into a tool for physicians in limited areas of medicine. Dr. E.H. Shortliffe will again take over the leadership of the project and intends to become principal investigator on new grant applications that we submit.

Because of our difficulty in securing funding for a phase of the project which is neither basic research nor immediate health care delivery, we may be forced to give up the idea of moving MYCIN out of the research environment with federal support. There are many challenging research questions left to be explored both in artificial intelligence and in medical computing, while we seek a solution to the problem of funding development of health care technology, as opposed to either research or application.